UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/581,862	01/25/2007	Ottmar Gehring	095309.57760US	3922

23911 7590 04/04/2008
CROWELL & MORING LLP
INTELLECTUAL PROPERTY GROUP
P.O. BOX 14300
WASHINGTON, DC 20044-4300

EXAMINER	
KIM, EDWARD J	
ART UNIT	PAPER NUMBER
2155	
MAIL DATE	DELIVERY MODE
04/04/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/581,862	GEHRING ET AL.	
	Examiner	Art Unit	
	EDWARD J. KIM	2155	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *25 January 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-9* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☐ Claim(s) _____ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *25 January 2007* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☒ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *06/05/2006*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is a responsive to the application filed on 01/25/2007.

2. Claims 1-9 are pending in this office action.

3. The claims are directed towards a method for loading a software module into a processor unit in a controller, optimizing the processor utilization level in networked controllers.

Drawings

4. The drawings are objected to because the unlabeled rectangular box(es) shown in the drawings should be provided with descriptive text labels. Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Claim Objections

5. Claim 1 and 8 are objected to because of the following informalities:

Claim 1 recites, "running sending" in the second paragraph. Appropriate correction is required.

Claim 8 recites, "data are sent" in the last paragraph. Appropriate correction is required.

Claim Rejections - 35 USC § 112

6. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

7. Claims 1, 8, and 9 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Regarding claim 1, it is vague and indefinite to what component of the invention carries out the method claimed in claim 1. It is unclear to what component selects the controller (as described in step ii)), what checks cyclically (as described in the third paragraph), and what determines which of the controllers (as described in the fourth paragraph), failing to particularly point out and distinctly claim the subject matter which the Applicant regards as the invention.

Claim 1 recites, "a software module in a controller on which said software is running sending to the data bus". Due to the structure of the claim language, it is unclear to what the claimed subject matter is. Appropriate grammatical correction is required to clarify and particularly point out the claimed subject matter.

Claim 1 recites, "…an appropriate identifier containing information indicating its operating status…" It is unclear to what "its" is referring to the software module or the controller. Appropriate correction is required to clarify and particularly point out the claimed subject matter.

Claim 1 recites, "wherein said determining step is made by virtue of the controllers involved in sending..." The term "made by virtue" is vague and indefinite to what the limitation is referring to and how the determining step is performed. Therefore, the claim fails to particularly point out and distinctly claim the subject matter which the Applicant regards as the invention.

Regarding claim 8, it is disclosed in paragraph [0040] of application (refer to the publication of the application), that the controllers may also perform a plurality of primary tasks. The limitation, "primary control task" is vague and indefinite to what the term is referring to, failing to particularly point out and distinctly claim the subject matter which the Applicant regards as the invention.

In the same context, claim 8 recites "subsidiary task" which is not defined in the disclosure by the Applicant, failing to particularly point out and distinctly claim the subject matter which the Applicant regards as the invention.

Regarding claim 9, it is unclear to what component of the system carries out the method claimed. It is unclear to what checks the data bus for an identifier (as described in the fourth paragraph and seventh), what determines which controller has the greatest available capacity (as described in the fifth paragraph).

Deleted: Page

8. Insufficient Antecedent Basis: Claim 8 recites the limitation "primary task" in the sixth paragraph. There is insufficient antecedent basis for this limitation in the claim.

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. Claim 1-3, and 5-9 rejected under 35 U.S.C. 103(a) as being unpatentable over Kayano et al. (US Patent #5,544,054), hereinafter referred to as Kayano, in view of Chavez (US Patent #7,308,499 B2).

Kayano discloses a vehicle multi-processor control system and method with processing load optimization, where the plurality of control processors are connected by LAN, BUS and so on.

Regarding claim 1, Kayano discloses, a method for operating a software module on a processor unit in a controller networked via a data bus in a vehicle (Kayano, Abstract), wherein i) the software module is executable in a plurality of controllers which interchange data via the data bus (Kayano, Abstract, col.1 ln.42-63, col.1 ln.55-56, col.1 ln.61-65) , ii) selection of the controller on which the software module is operated is made based on the available computational capacity of the controllers which are currently in operation (Kayano, Abstract, col.1 ln.42-63, col.1 ln.44-49), and iii) each of the controllers can turn off the software module when a utilization level of its processor is high, and as soon as the software module has been

turned off, the software module is to be started again on another controller (Kayano, col.3 ln.45-54, col.11 ln.14-16, col. 7 ln.40-45, col.4 ln.62-67. Software modules are terminated according to the computing capacity and transferred to run on a different processor.): said method comprising:

a software module in a controller on which said software module is running sending to the data bus, either cyclically or upon request, an appropriate identifier containing information indicating its operating status and the identity of the controller on which it is running (Kayano, col.1 ln.42-62, col.2 ln.65 – col.3 ln.12, col.3 ln.40-47, col.6 ln.40-47. Kayano discloses a method and system where software modules are transferred to be run on processors with less load or in other words, less utilization rate. The system is disclosed to be constantly acknowledging changes in the load state and adjusts accordingly.);

checking cyclically to determine whether and on which controller the software module is running, based on said identifier (Kayano, col.1 ln.42-62, col.2 ln.65 – col.3 ln.12, col.3 ln.40-47, col.6 ln.40-47.);

wherein said determining step is made by virtue of the controllers involved sending in rotation or by means of a request, information that is indicative of their available computational capacity (Kayano, col.1 ln.42-62, col.2 ln.65 – col.3 ln.12, col.3 ln.40-47, col.6 ln.40-47.).

Kayano discloses that the processor with lower load, greater capacity available, is chosen for executing the software (Kayano, col.6 ln.40-49). It was common and well-known in the art of at the time the invention was made to select a processor which has the greatest free computation capacity for executing a software module, as shown by Chavez (Chavez, col.3 ln.4-15. Chavez discloses a method for dynamic load balancing in a network, where the processor

with the greatest free computation capacity is chosen.). Processor clock frequencies are commonly used in the art to distinguish processor types. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Kayano to select the processor with the greatest computation capacity for executing the software module.

Regarding claim 2, Kayano and Chavez disclosed the limitations, as described in claim 1, and further discloses, a method wherein before the software module is executed it is ascertained which of the controllers provides the maximum free computation capacity and the software module is started on the determined controller (Kayano, col.3 ln.45-54, col.11 ln.14-16, col. 7 ln.40-45, col.4 ln.62-67. Software modules are terminated according to the computing capacity and transferred to run on a different processor. Chavez, col.3 ln.4-15. Chavez discloses a method for dynamic load balancing in a network, where the processor with the greatest free computation capacity is chosen.).

Regarding claim 3, Kayano disclosed the limitations, as described in claim 1, and further discloses, a method wherein the controller on which the software module is running compares its computation capacity with the computation capacity of the other controllers and either continues to operate or terminates operation of the software module based on the comparison (Kayano, col.3 ln.45-54, col.11 ln.14-16, col. 7 ln.40-45, col.4 ln.62-67. Software modules are terminated according to the computing capacity and transferred to run on a different processor.).

Regarding claim 5, Kayano and Chavez disclosed the limitations, as described in claim 1, and further discloses, a method wherein the software module is started on a controller having the maximum free computation capacity (Kayano, col.3 ln.45-54, col.11 ln.14-16, col. 7 ln.40-45,

col.4 ln.62-67. Software modules are terminated according to the computing capacity and transferred to run on a different processor. Chavez, col.3 ln.4-15. Chavez discloses a method for dynamic load balancing in a network, where the processor with the greatest free computation capacity is chosen.).

Regarding claim 6, Kayano disclosed the limitations, as described in claim 1, and further discloses, a method wherein the software module is stored in a memory in the controllers (Kayano, col.1 ln.60-63, col.3 ln.51-67).

Regarding claim 7, Kayano disclosed the limitations, as described in claim 1, and further discloses, a method wherein: an identifier for the software module is sent to the data bus cyclically or upon request; and the identifier contains information about an operating state and the operating controller of the software module (Kayano, col.1 ln.42-62, col.2 ln.65 – col.3 ln.12, col.3 ln.40-47, col.6 ln.40-47. Kayano discloses a method and system where software modules are transferred to be run on processors with less load or in other words, less utilization rate. The system is disclosed to be constantly acknowledging changes in the load state and adjusts accordingly.).

Regarding claim 8, Kayano discloses, a networked controller having software modules stored in a controller's memory; wherein:

the software modules perform primary control tasks; a software module with a subsidiary task can be additionally stored in a microcontroller's memory by the controllers (Kayano, Abstract, col.1 ln.42-65, col.2 ln.65 – col.3 ln.12, col.3 ln.40-47, col.6 ln.40-47);

the controllers have process cycles; a process cycle is terminated after a particular time has elapsed, the data ascertained in the process are output onto the data bus, and the process

cycle is started again; the process cycle for the controllers is determined by the software modules for one of the primary task, the operating system and a bus protocol; and when a process cycle or a process cycle time has elapsed, data are sent to the data bus which characterize their current processor utilization level and processor type used, with the controllers using these data to ascertain the utilization level of the other controllers.

In regards to the above limitations set forth by the claim, Kayano discloses a method and system where software modules are transferred to be run on processors with less load or in other words, less utilization rate. The system is disclosed to be constantly and regularly acknowledging changes in the load state and adjusts accordingly (Kayano, col.1 ln.42-62, col.2 ln.65 – col.3 ln.12, col.3 ln.40-47, col.6 ln.40-47.). Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have set intervals, such as process cycles to output, check, and determine the status of each controller.

Regarding claim 9, Kayano discloses, a method of operating a network of controllers which are coupled via a data bus, each of which controllers has at least one processor, and has installed thereon the same software module which can be executed by the processor contained in any one of the controllers (Kayano, Abstract, col.1 ln.42-63, col.1 ln.55-56, col.1 ln.61-65), each of said controllers being configured such that it can turn off the software module when a utilization of its processor is high (Kayano, col.3 ln.45-54, col.11 ln.14-16, col. 7 ln.40-45, col.4 ln.62-67, col.6 ln.40-49. Software modules are terminated according to the computing capacity and transferred to run on a different processor.), said method comprising: each controller sending via the data bus, information regarding a current utilization level of its at least one processor (Kayano, col.1 ln.42-62, col.2 ln.65 – col.3 ln.12, col.3 ln.40-47, col.6 ln.40-47.);

whenever said software module is running in a particular one of said controllers, said software module in said particular controller sending via the data bus, an identifier indicating its operating state and identifying the particular controller (Kayano, col.1 ln.42-62, col.2 ln.65 – col.3 ln.12, col.3 ln.40-47, col.6 ln.40-47.);

checking said data bus to determine whether an identifier is present; if no identifier is found in said checking step, determining which of the controllers has the greatest available computation capacity, based on its current utilization level as sent via the data bus (Kayano, col.1 ln.42-62, col.2 ln.65 – col.3 ln.12, col.3 ln.40-47, col.6 ln.40-47. Kayano discloses a method and system where software modules are transferred to be run on processors with less load or in other words, less utilization rate. The system is disclosed to be constantly acknowledging changes in the load state and adjusts accordingly.);

said controller with said greatest available utilization level starting operation of said software module, and said software module sending to said data bus, an identifier indicating its operating status and the identity of the controller in which it is running (Kayano, col.1 ln.42-62, col.2 ln.65 – col.3 ln.12, col.3 ln.40-47, col.6 ln.40-47);

if an identifier is present on the data bus in said checking step, the controller on which said software module is running ascertaining its own processor utilization level and comparing its computation capacity with the available computational capacity of other controllers coupled via the data bus (Kayano, col.1 ln.42-62, col.2 ln.65 – col.3 ln.12, col.3 ln.40-47, col.6 ln.40-47);

if the utilization level of the controller on which the software module is greater than that of one of said other controllers, said controller on which said software module is running ceasing operation of said software module; and said one of said other controllers starting operation of

said software module, and said software module sending to said data bus an identifier indicating that it is running and identifying said one controller (Kayano, col.3 ln.45-54, col.11 ln.14-16, col. 7 ln.40-45, col.4 ln.62-67. Software modules are terminated according to the computing capacity and transferred to run on a different processor.).

Kayano discloses that the processor with lower load, greater capacity available, is chosen for executing the software (Kayano, col.6 ln.40-49). It was common and well-known in the art of at the time the invention was made to select a processor which has the greatest free computation capacity for executing a software module, as shown by Chavez (Chavez, col.3 ln.4-15. Chavez discloses a method for dynamic load balancing in a network, where the processor with the greatest free computation capacity is chosen.). Processor clock frequencies are commonly used in the art to distinguish processor types. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Kayano to select the processor with the greatest computation capacity for executing the software module.

11. Claim 4 is rejected under 35 U.S.C. 103(a) as being unpatentable over Kayano et al. (US Patent #5,544,054), hereinafter referred to as Kayano, in view of Chavez (US Patent #7,308,499 B2), in further view of Official Notice taken by the Examiner.

Regarding claim 4, Kayano disclosed the limitations, as described in claim 1, and further discloses, a method wherein the computation capacity of a controller is ascertained from the processor utilization level and processor type (Kayano, co.1 ln.42-62, col.3 ln.2-4.).

Although Kayano does not explicitly state that the processor type is also considered in the computing the capacity of a controller, "official notice" is taken by the Examiner that this method was well-known in the art at the time the invention was made. It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the teachings of Kayano to take into account the processor type for computing the capacity of a processor. One would have been motivated to do so since it was well-known and common in the art that processor type also determines the computing capacity.

Conclusion

Examiner's Note: Examiner has cited particular columns and line numbers in the references applied to the claims above for the convenience of the applicant. Although the specified citations are representative of the teachings of the art and are applied to specific limitations within the individual claim, other passages and figures may apply as well. It is respectfully requested from the applicant in preparing responses, to fully consider the references in entirety as potentially teaching all or part of the claimed invention, as well as the context of the passage as taught by the prior art or disclosed by the Examiner.

In the case of amending the claimed invention, Applicant is respectfully requested to indicate the portion(s) of the specification which dictate(s) the structure relied on for proper interpretation and also to verify and ascertain the metes and bounds of the claimed invention.

The prior art made of record and not relied up on is considered pertinent to applicant's disclosure.

- Uematsu, US Patent #6,606,658 B1, Apparatus and method for server resource usage display by comparison of resource benchmarks to determine available performance.
- Lele, US Patent #7,181,524 B1, Method and apparatus for balancing a load among plurality of servers in a computer system.
- Banerjee, US Patent #6,816,510 B1, Method for clock synchronization between nodes in a packet network.
- Primak et al., US Publication #2002/0010783 A1, System and method for enhancing operation of a web server cluster, discloses a distributed system and method for balancing connection load among servers.
- Lea et al., US Patent #6,314,447 B1, System uses local registry and load balancing procedure for identifying processing capabilities of a remote device to perform a processing task.
- Logston et al., US Publication #2002/0032754 A1, Method and apparatus for profiling in a distributed application environment, discloses an improved method and apparatus for deriving and characterizing the resource capabilities of client devices in a distributed application network environment.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Edward J. Kim whose telephone number is (571) 270-3228. The examiner can normally be reached on Monday - Friday 7:30am - 5:00pm EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Saleh Najjar can be reached on (571) 272-4006. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Edward J Kim/
Patent Examiner, Art Unit 2155

/saleh najjar/
Supervisory Patent Examiner, Art Unit 2155